Component based approach using OMNeT++ for Train Communication Modeling

Juan-Carlos Maureira, Paula Uribe, Olivier Dalle, Jorge Anaya, Takeshi Asahi

Component based approach using OMNeT++ for Train Communication Modeling

Juan-Carlos Maureira[‡], Paula Uribe[‡]
INRIA - Sophia Antipolis
Email: {jcmaurei, puribejo}@sophia.inria.fr

Olivier Dalle
INRIA - Sophia Antipolis
Email: odalle@sophia.inria.fr

Takeshi Asahi[†], Jorge Amaya[†]
Center for Mathematical Modeling
Universidad de Chile
Email: {tasahi, jamaya}@dim.uchile.cl

***Abstract*—This paper reports on our experience in using OMNeT++ to develop a network simulator focused on railway environments. Common design problems are analyzed, making emphasis on radio communication models. Scalability issues are raised when modeling the large topologies that are associated with railway communications. Our conclusions point out that model reusability must be reinforced and that a component-based design must be adopted in order to build a tool for generating valuable performance results.**

***Index Terms*—OMNeT++, discrete event simulation, modeling, train communication**

I. Introduction

Building network simulations is a complex problem because networks involve numerous layers and technologies, and taking into account all this complexity in simulation models often results in too much computational complexity. Abstraction is a common process to reduce the system complexity, enabling an easier implementation of the model and reducing computation costs. Nevertheless, the risk of this practice is to neglect sensitive details that may drive researchers to misleading conclusions. Therefore, a difficult problem is to decide how much abstraction is required to obtain usable results from a network simulated model. In a railway context, where wireless communication could be used to provide train to ground connectivity, this problem is even more difficult, since wireless communication is more complex to be modeled in simulations. In this paper, we report on our experience and conclusions about the requirements when developing a network simulator focused on railway environments. Our discussion is based on the experience we gained developing network simulations in the context of the *InteGRail* project, using the OMNeT++ simulator. While this network simulator is useful to get a rough idea about railways communications with several networking technologies, we realize that, to obtain valid results using simulation, a more careful approach must be used. In this work, we evaluate the relevance of using a particular network modeling framework, the *INET* Framework, designed atop the OMNeT++ simulator. This framework provides wired and wireless models, as well as higher level protocols such as IP, TCP and UDP. We review its characteristics in order to evaluate all the required elements to model a railway scenario, in terms of radio communications, protocols, scalability and mobility.

This paper is organized as follows: in the next Section, we provide a short description of OMNeT++ and we describe its companion *INET* network modeling framework. In Section III we analyze the railway networking scenario within the simulation context, putting emphasis on the radio model requirements needed to obtain valid results; we also analyze scalability issues when large topologies are simulated, with special emphasis in mobility issues. In Section IV, we summarize our first experience in building a simulation model focused on railway scenarios and, finally, in Section V, we contrast our first experience against our current experience in simulating networking system in railway scenarios, converging to this component based approach to model train communication systems.

II. OMNeT++ and Network Models

In this section, we briefly describe the OMNeT++ simulation framework. Following, we discuss the *INET* network modeling framework and its main features. In particular, we discuss the *INET* current validation status, based on the literature and our own experience.

A. The OMNeT++ Simulator

OMNeT++ [1] is a C++ based discrete event simulator designed to model communication networks and distributed systems. It allows to simulate models in which the entities communicate with each other by means of messages. These entities are implemented by means of component and may have a hierarchical structure. The model of the system is specified using a proprietary, rich featured, architecture description language, called NED. This language allows the specification of a system in terms of simple modules (atomic systems), compound modules (composed by other atomic/compound modules), and a set of gates/links handling the communications between these modules. The communication between modules is ruled by a channel model, allowing a customizable representation of the message exchange. Additionally, it provides an extensible instrumentation of atomic modules and channels, allowing the gathering of data (series of measurements-vectors and scalars values - metrics) during the simulation. The behaviour of atomic modules and channels is implemented in

[‡]On leave from CMM - Universidad de Chile.

[†]These co-authors have been supported by FONDAP and by Basal CONICYT-Chile Program.

This work has been partially done as a part of InteGRail, Project PL 012526, European Commission, VI Framework Program.

C++ by extending the OMNeT++ object hierarchy class tree. It supports experiment setup and replication. The Random Number Generator (RNG) is also customizable, having as default the *Mersenne Twister* RNG, which uses the MT19937 RNG by Makoto Matsumoto[2]. The *Linear Congruential Generator* (LCG32)[3] is also available. OMNeT++ also provides an optimistic parallel simulation[4] kernel, allowing to scale the simulation to large models by means of partitioning the system (the set of components) along different CPUs. It is worth to mention that to simulate a system in a parallel way, the RNG must be revisited in order to generate a unique sequence of random numbers for all the processors[5].

B. Networking Models

OMNeT++ has gained popularity in the last few years when studying networking problems. A variety of public-domain models have emerged from this popularity: the *INET* Framework, MiXiM[6] and the Mobility Framework (MF), just to mention a few. Currently, the *INET* Framework is the only networking model, from the models built on top of OMNeT++, that provides several high-level networking protocols (Applications, IP, TCP, UDP, among others), in addition to the wired and wireless physical layers provided by MiXim and MF as well. The results delivered by the high-level protocols and wired physical layer seems to be reasonable when compared to simple real experimentation. However, their validity still needs to be further evaluated for complex models. Regarding wireless physical layer model, a heedful review must be done. When modeling radio communications in simulation, several false axioms are usually assumed [7]. Postulates such as *the world has a flat surface*, *the propagation radius is circular* and *the communication is symmetrical* are some of them. Nevertheless, some of these postulates may affect simulation results more than others, depending on how the propagation, interference and reception models reflect them. *INET*'s interference model is represented by an additive model, considering all the on-going transmissions to calculate the SNR along the packet reception. Thus, an SNR vector is evaluated to determine when a packet (or part of it) is lost due to a low SNR. The *INET*'s default propagation model is the classical *Free Space Pathloss* model, but the *Two-ray* (with ground reflection) model is also included. Additionally, Zitterbart et al.[8] have introduced probabilistic propagation models, but into the Mobility Framework model, not in the *INET* model. However, these models could be adjusted to be used in *INET* as well. Regarding the *INET*'s propagation model, it assumes a circular propagation area and symmetrical communications[1] by default. But, nowadays, there are some *INET*'s branches that are already supporting asymmetric communication and interference-dependent coverage area. An important element of the *INET*'s radio model is the *Channel Controller* module. This module is an abstraction of the radio channel communication, being in charge of representing the radio propagation between nodes, providing all the required information for radio devices (PHY module) to use the interference and reception model, and keeping records of all the on-going transmissions in a certain time.

There are not many studies in the current literature[2] that focus on the validation of *INET* radio models. In [9], the authors present an empirical validation of the MF radio model in the context of Wireless Sensor Networks (WSN). In [10], the author compares the accuracy of the MF 802.11 radio model against Bianchi's theoretical model[11]. Nevertheless, we highlight the work presented by Bredel and Bergner[12] on the accuracy of the IEEE 802.11g radio model in *INET*. They drove an extensive measurement study on a highly controlled environment (almost interference free), measuring metrics like throughput, delay and packet inter-arrival time. They contrasted these metrics against similar ones obtained from simulating the same scenario. They pointed out that results on throughput and delay were statistically similar for long observation times in most of the cases they studied. But, they noticed a difference on the packet inter-arrival time distribution, evidencing differences on the packet scheduling policy. These differences could invalidate studies where this kind of metric is relevant, such as rare event problems, or short observation times. Finally, regarding the interference model accuracy, we can mention the work presented in[13], where the authors attempt to improve the *INET*'s additive interference model by including real traces of wireless packets in order to generate an *interfering background traffic*, achieving an interaction between the simulated system and an external scenario, characterized by the packet traces.

Summarizing, in our experience, the *INET* Framework provides a reliable and extensible implementation of applications and protocols, reasonably modeled down-to the data-link layer. However, we acknowledge that the physical layer, specially radio models, could (and should) be always been questioned due to its level of abstraction in radio propagation, interference and reception models. In this matter, we can say that the validation of the *INET* Framework is still an open issue. However, the previously presented works are a good signal of how reasonable are the results produced by the *INET* Framework when simulating long periods of time, such as a train trip. Additionally, it is also reasonable to consider real interfering scenarios by including packet traces into the simulated scenario.

III. Networking in a Railway Scenario

Within a railway scenario, we define "networking" as the need to exchange traffic between an in-motion network (on-board the train) and a fixed infrastructure network. The Fig.1 helps to depict the scenario.

Several ways to address this communication issue are proposed: Satellites, UMTS 3G, Wifi, WiMax, or a combination of them, where each solution presents its corresponding advantages and disadvantages. Studying such solutions experimentally is feasible, but not as easy as simulations.

[1] "If you hear me, I can hear you"

[2] Reviewed until August 2009

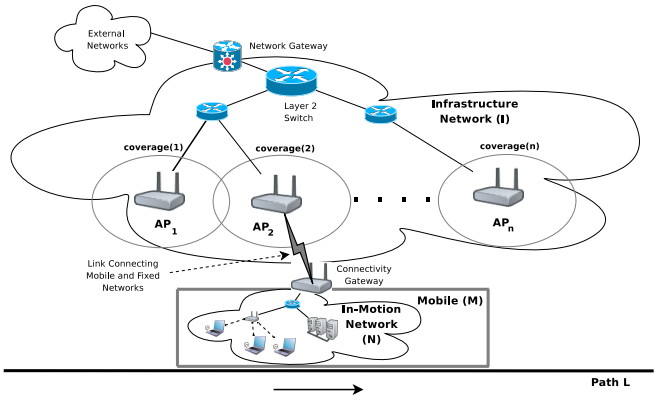

Fig. 1. The Railway Scenario

The most common networking problem in this scenario is the so-called *Handover Problem*. In the Fig.1, the gateway device placed on-board the mobile M must switch the link from one $AP(i)$ to the next one while M is traveling along the path L. When a single technology is used, this problem is known as *Horizontal Handover*, while between different technologies, it is called *Vertical Handover*. In the following, we analyze the radio model, mobility issues, and scalability constraints of *INET* when studyng these kinds of problems in a railway scenario.

A. Radio Models

Radio models are usually described in terms of three main components: propagation, interference and reception models. Modeling accurately all of them is computationally too expensive and, on the other hand, an over-simplified model may lead to misleading conclusions based on non-realistic representations. As stated in the previous section, the *INET* networking model provides a basic and extensible radio model, which is composed by: two propagation models (the *Free Space Pathloss* model and the *Two-Ray* model); an additive interference model (capable of handling collisions by adding the transmission power of all the coexisting wireless communications to the received power); and a reception model (capable of calculating the BER according to the BPSK, 16-QAM and 256-QAM modulation techniques). The validation of these models was already discussed in the II-B section.

In order to describe in more detail the network within a railway simulated scenario, we first list some of the main characteristics:

- Large simulation times (Train trips usually take hours).
- Large geographical areas are traversed by the train;
- The in-motion network (on-board) connects to the fixed infrastructure network through a wireless link;
- Usually, the train travels at high-speed (over 120 Km/h);
- The infrastructure network is placed close to the train path and is assumed to be internally connected by a combination of wireless and wired links, depending on the distance to the backbone nodes;
- Traffic exchange between the in-motion network and the external networks is bi-directional;
- Metrics commonly used are the overall system throughput, end-to-end delay and connection continuity.

In this context, since the studied metrics are related to the overall system performance, our experience told us that every possible error induced by the physical radio models is negligible compared to the potential misestimations caused by the *Media Access Control Methods* (MAC), handover algorithms or even the infrastructure network topology. Therefore, as a first step we propose to focus the design and the implementation of the simulator on a more accurate description of the MAC and the handover algorithms, instead of implementing an extremely detailed radio physical layer model, such as ray-tracing methods[14].

Towards the analysis of our approach to radio modeling, we will refer to the IEEE 802.11 radio technology, focusing on interference and mobility issues. For this case of study, there are two possible scenarios: urban, where the interference is high and the mobility is low, and sub-urban, where the interference is low and the mobility is high. In the first scenario, real traces of interfering traffic can be injected to the simulation[13], so as to achieve a better model reflecting reality. Also, the *Free Space Pathloss* model is able to describe the radio propagation properly, since the IEEE 802.11 Access Points are assumed to be located close to the railway, avoiding obstacles between the train radio and the infrastructure. On the second scenario, studies presented in [15], [16] and [17] have shown successful wireless connections with a single AP, when the mobile is traveling up-to 180 Km/h, without modifying the IEEE 802.11 protocol. Thus, the only issue is the handover frequency at high-speed, that is currently being studied.

Finally, we emphasize that buggy MACs or handover algorithms could produce bigger errors on simulation results than radio models.

B. Mobility

Train Mobility is one of the easiest issues to solve in OMNeT++. However, it must be analyzed carefully, since it does not follow exactly the same constructors as wireless host mobility. Wireless mobility is normally related to providing movement (linear, constant speed, circular, grid, etc.) to a host that contains wireless devices. Obviously, a parallel can be established between the host and the train, since both have wireless devices communicating with a fixed infrastructure network. But, wireless mobility elements by themselves are not sufficient to describe a train mobility in all its nature. Elements such as variable speed (including full stops), intersections (when several routes are present), and railway network topologies are required to depict completely a railway scenario.

We define train mobility as the description of a train route within a *Railway Network*. The *Railway Network* is composed by a set of single railway paths connected each other by means of *intersections*. Along each single railway path, the train must be able to change its *speed*, or even make a full stop

for a certain duration (i.e. at a train station). Additionally, at an intersection, the train must be able to decide which path it must follow in order to follow its route. Thus, all these concepts must be reflected into the simulated scenario in a comprehensible and scalable way.

We propose two new modules within the simulated railway scenario: the *RailwayScenarioManager* and the *RailwayMobility*. The first one is in charge of describing the railway network, providing the intersections and railway topology. This module is placed inside the simulated playground (similar to the *ChannelController*), since its operation is related to the whole simulation scope. On the other hand, the *RailwayMobility* module is placed inside each train, being in charge of moving the train along the railway and processing pre-defined events such as the variation of the speed or the decision of the route. Figure 2 depicts the placement of these modules within the simulation scenario.

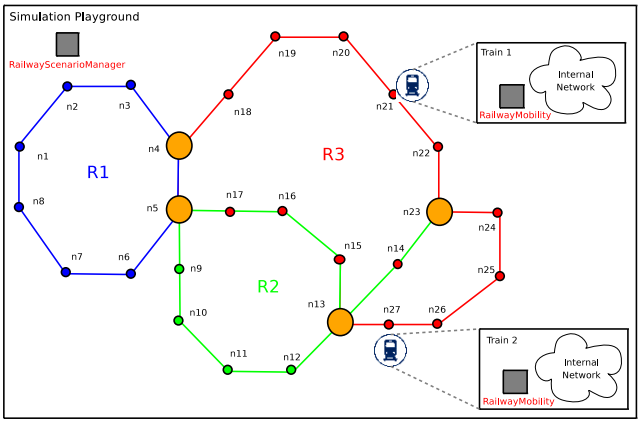

Fig. 2. RailwayScenarioManager and RailwayMobility placement within the simulated scenario.

The *RailwayScenarioManager* builds a railway network representation based on an input XML-file defining each railway path by means of geo-referenced, unique-id nodes sequence (as is shown in the figure 2). Intersections are described using common node-ids in multiple single paths. It is worth to mention that a node requires to be geo-referenced only one time. The *RailwayMobility* Module uses the same XML-file to define the starting point of the train on the network and a sequence of events affecting the train route, such as speed changes or routing decisions. The input XML-file looks like this:

```
<Railway id="R1">
  <Node id="n1" x="..." y="..." linkTo="n2">
  ...
  <Node id="n4" x="..." y="..." linkTo="n5">
  ...
  <Node id="n8" x="..." y="..." linkTo="n1>
</Railway>
<Railway id="R2">
  <Node id="n9" x="..." y="..." linkTo="n10">
  ...
  <Node id="n13" x="..." y="..." linkTo="n14" linkTo="n15">
  ...
  <Node id="n14" x="..." y="..." linkTo="n23">
  <Node id="n17" x="..." y="..." linkTo="n5">
</Railway>
```

```
<Railway id="R3">
  <Node id="n4" linkTo="n18">
  ...
  <Node id="n23" x="..." y="..." linkTo="n24">
  ...
  <Node id="n27" x="..." y="..." linkTo="n13">
</Railway>
...
<Train id="train1">
  <Start node="n21">
  <Events loop="true">
    <Event t="..." speed="...">
    <Event node="n23" nextNode="n24">
    <Event node="n25" speed="0" wait="10min">
    <Event node="n25" speed="40" acceleration="10">
  <Events>
  <End node="n5>
</Train>
<Train id="train2">
  <Start node="n27">
  <Events>
  ...
  </Events>
  <End node="n1">
</Train>
```

Railways can be defined as a set of nodes, connected by a set of links (similar to a Directed Graph). Each node has the *linkTo* attribute, which indicates the next node and therefore, the railway *direction*. I.e., a train coming from node $n14$ to node $n23$, can only follow the $n24$ direction, since $R3$ defines an unique direction from $n23$ to $n24$. A train coming from $n27$ to $n13$, instead, can decide between the $n14$ and $n15$ directions.

Each train defines a *starting/ending node*, and a set of events to be triggered at a certain time (**t** attribute), or when a certain node is visited (*node* attribute). Each event can set several parameters at the same time, such as *speed* and *acceleration*. There are some train actions that depend on more than one parameter. For example, if we want to indicate a train stop, both, the *speed* and the *wait* time parameters must be specified ($speed = 0$ and $wait = 10min$, for instance). Otherwise, if only the *speed* parameter is specified, the train will remain on that node forever.

Events can define *cycles*, which means that when all the defined events have been triggered, the train can start again, allowing the definition of loops in the train route.

Summarizing, the defined modules and the description language of scenarios and train events, in conjunction with the mobility features of OMNeT++ can describe fully a railway scenario, with multiple trains/railway networks in an intuitive easy way.

C. Scalability and Large Topologies

Simulated railway scenarios normally involve large topologies and a per-packet detail in order to analyze the overall system performance. These two issues, in conjunction with a fine mobility (in a meter resolution), requires special attention when designing a simulator. The amount of events to be handled by the simulation core and the amount of memory involved in representing thousands of objects may lead the simulation study to the infeasibility. In this direction, we analyze the capabilities of OMNeT++ of handling large simulations.

As we mentioned in the section II-A, OMNeT++ is equipped with a parallel simulation kernel. This feature allows to split a large model into "partitions" in order to assign them to different processors. The Fig.3 illustrates the model partitioning. All the links between modules that reside in different partitions (red dashed links) are implemented by OMNeT++ in MPI, providing an efficient way to communicate modules across CPUs.

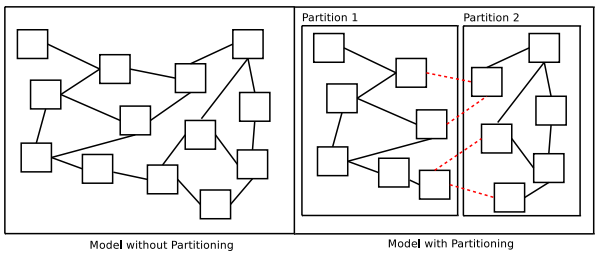

Fig. 3. Model Partitioning

It is worth to mention three important aspects in model partitioning. One aspect is when parallel simulation is used, a synchronization algorithm must be selected in order to keep synchronized each partition in time, and avoid to execute events in a wrong sequence of time. On this, OMNeT++ provides the Null-Message-Algorithm to synchronize the Future Event Set on each CPU. This is an optimistic algorithm, and it is discussed with more detail in [18]. The other aspect is how to perform partitioning. The natural option is a per-node partition (as is depicted in the Fig.3), but a rol-based partitioning approach can also be used. This approach sets in the same partition all the components that need to be communicated more often, so that, they will use the same RNG to perform randomization of replicas. In this way, we state the third aspect, the RNG in parallel simulations. As we mentioned earlier [5], simulation is about randomization of experiments to be able to study a factor effect on the system. When a single simulation cannot be exactly reproduced with the same RNGs setting, the simulation is not valid. So, when parallel simulation is used, this issue is not easy to solve. Thus, role-based partitioning could be a way to address this problem.

Summarizing, large topologies can be simulated and executed in OMNeT++ within a human time-scale. It is worth to mention that a parallel simulation is not focused on performance, but on making the experiment runnable.

IV. Our first Implemented Model

Under the *InteGRail* project (www.integrail.info), a basic railway communication model was implemented. The main objective of this first implementation was to evaluate in a coarse mode various communication architectures, technologies and algorithms related to railway communication (train to ground and vice verse), without dealing directly with issues like detailed network stack on each particular technology. The simulator design and implementation was focused on providing a very simple and parametric radio model, in order to emulate radio features implemented on back-end protocols, such as GSM or WiFi by providing a set of parameters and Media Access Methods (TDMA, FTDMA, CDMA), duplexing techniques, Doppler effect, fading channels, multi-Path packet receiving. The simulator was designed to test several network selector algorithms in a multi-technology environment on-board the train, considering parameters such as reception signal and delay to the infrastructure. The simulator is functional, although there is no validation of the radio model and it does not consider an interference model. Thus, when evaluating its results on testing a network selector algorithm, we realized that a more accurate modeling of higher protocols such as IP, TCP or UDP is more relevant than the lower physical layers in order to have better results on metrics that the network selector required. Additionally, we realized that the vertical and horizontal handover considered in this simulator may not be sufficiently accurate, since accessing times in GSM networks and OSI layer 3 addressing delays are completely ignored.

In summary, this experience showed us the direction to take when designing simulation models focused on railways environment. Throughout this development, the priorities on the important implementation aspects were clarified when searching a continuous connectivity between an in-motion network and a fixed one. Further efforts, specially when using the *INET* Framework have reported us more usable results when studying these sort of problems.

V. Conclusions and Further Work

This paper reports on our experience in using OMNeT++ to develop a network simulator focused on railway environments. While we think that it is very usable, our conclusions can not be extended when using other simulators. Thus, our conclusions are two-fold: on one side, what we learned from experience in terms of simulator design; and on the other side, how we can exploit this experience for future designs of network simulators in railway communication studies.

Our first experience on designing a network simulator focused on railway environments has shown several important points to consider. First, the simulation tool must provide various abstraction levels in order to address the model from multiple points of view. Second, radio models are important, but errors induced by them might not be so important, compared to those induced by higher level algorithms, such as MAC algorithms, handover algorithms or transport protocols. Third, to represent the train mobility, extra elements are required in order to depict correctly common train behaviours, such as train routes, speed changes, stops and intersections. Fourth, scalability of the model is important to simulate large scenarios over long simulated times and to generate representative results of the overall system performance.

Towards the accomplishment of all these points, we have realized that the **reusability** of models is a requirement. Implementing all the *OSI* networking layers in order to have the required elements to properly describe a railway scenario is an enormous effort, in terms of time and experimentation.

Model reusability could help us to provide an already validated network model, and to extend it, to implement the extra elements required to model our scenario. This way, we can focus on modeling the communication within a railway scenario, rather than on having a suitable networking model.

Going further in this direction, we realized that OMNeT++ provides all the simulation constructors to properly support the model **abstraction** and **scalability** issues above mentioned. Furthermore, the *INET* networking model is robust and extensible enough to allow us to implement all the extra elements and to replace simulation components if needed in order to build a simulator centered on a railway scenario.

This component-based approach, in conjunction with OMNeT++ and *INET*, has being used in other studies related to railway scenarios, achieving good results, from the model development and use points of view, when measuring the overall system's performance.

Some additional work can be done by studying the model partitioning when large scenarios are involved. We propose that the role-based partition approach **would** be reasonable when solving the partitioning, but the RNG issues and the cost of the communication between a large number of distributed CPUs is clearly a field to address as further work.

VI. Acknowledgments

The authors would like to thank everyone who helped us to write this paper. This work was partly founded by the IST-FET AEOLUS project and CONICYT Chile.

References

[1] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," in *Simutools '08: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, ICST, Brussels, Belgium, Belgium, 2008, pp. 1–10. [Online]. Available: http://portal.acm.org/citation.cfm?id=1416222.1416290

[2] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Trans. Model. Comput. Simul.*, vol. 8, no. 1, pp. 3–30, 1998.

[3] D. E. Knuth, *Art of Computer Programming, Volume 2: Seminumerical Algorithms (3rd Edition) (Art of Computer Programming Volume 2)*. Addison-Wesley Professional, November 1997.

[4] T. Phan and R. Bagrodia, "Optimistic simulation of parallel message-passing applications," in *PADS '01: Proceedings of the fifteenth workshop on Parallel and distributed simulation*. Washington, DC, USA: IEEE Computer Society, 2001, pp. 173–181.

[5] C. J. K. Tan, "On parallel pseudo-random number generation," in *ICCS '01: Proceedings of the International Conference on Computational Sciences-Part I*. London, UK: Springer-Verlag, 2001, pp. 589–596.

[6] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin, "Simulating wireless and mobile networks in omnet++ the mixim vision," in *Simutools '08: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 1–8.

[7] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," in *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*. New York, NY, USA: ACM, 2004, pp. 78–82.

[8] A. Kuntz, F. Schmidt-Eisenlohr, O. Graute, H. Hartenstein, and M. Zitterbart, "Introducing probabilistic radio propagation models in omnet++ mobility framework and cross validation check with ns-2," in *Simutools '08: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 1–7.

[9] U. M. Colesanti, C. Crociani, and A. Vitaletti, "On the accuracy of omnet++ in the wireless sensornetworks domain: simulation vs. testbed," in *PE-WASUN '07: Proceedings of the 4th ACM workshop on Performance evaluation of wireless ad hoc, sensor,and ubiquitous networks*. New York, NY, USA: ACM, 2007, pp. 25–31.

[10] B. Jang, "Wireless mac protocol design and analysis," PhD in Computer Science, INorth Carolina State University, 2009.

[11] G. Bianchi, L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the csma/ca mac protocol for 802.11 wireless lans," in *Personal, Indoor and Mobile Radio Communications, 1996. PIMRC'96., Seventh IEEE International Symposium on*, vol. 2, Oct 1996, pp. 392–396 vol.2.

[12] M. Bredel and M. Bergner, "On the accuracy of ieee 802.11g wireless lan simulations using omnet++," in *Simutools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, pp. 1–5.

[13] J.-C. Maureira, O. Dalle, and D. Dujovne, "Generation of realistic 802.11 interferences in the omnet++ inet framework based on real traffic measurements," in *Simutools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, pp. 1–8.

[14] F. Aguado Agelet, A. Formella, J. Hernando Rabanos, F. Isasi de Vicente, and F. Perez Fontan, "Efficient ray-tracing acceleration techniques for radio propagation modeling," *Vehicular Technology, IEEE Transactions on*, vol. 49, no. 6, pp. 2089–2104, Nov 2000.

[15] R. Gass, J. Scott, and C. Diot, "Measurements of in-motion 802.11 networking," in *Mobile Computing Systems and Applications, 2006. WMCSA '06. Proceedings. 7th IEEE Workshop on*, Aug. 2006, pp. 69–74.

[16] J. Ott and D. Kutscher, "Drive-thru internet: Ieee 802.11b for "automobile" users," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 1, March 2004, pp. –373.

[17] T. Tse, "Study of high-speed wireless data transmissions for railroad operation," Federal Railroad Administration - Office of Research and Development, Tech. Rep. RR07-10, April 2007.

[18] A. Varga, A. Y. Şekercioğlu, and G. K. Egan, "A Practical Efficiency Criterion for the Null-Message-Algorithm," in *Proceedings of European Simulation Symposium*, Delft, The Netherlands, 2003.